How the Web Functions

- When we hit https://www.techtonicgroup.com/ what happens? Don't focus too much on architecture (Monolithic, SOA, Microservices, etc.). Try to focus more on how the web functions.

When we hit https://www.techtonicgroup.com/ in our browser, we are telling our browser that we want to view the file stored at that location. The URL is a unique identifier that points to a specific computer or server where the file we want is stored. If we have the IP address where the computer or server storing the file we want is located then whatever device we are using (the client) can request to download information from that server. The URL is a nicer and easier way to get to the IP address of the computer where the files we want are stored. Where the actual IP address is a string of numbers, the URL https://www.techtonicgroup.com/ is like a nickname for that specific IP address so we will still get to the right address without having to remember the numbers. When we reach the correct IP address for the server that has the file we want, the client will request access to the file which in this case is a website.

- From start to finish how does that data reach you to be rendered in the browser?

Whatever device we happen to be using, and the computer or server that we are trying to get information from, are both connected via a data connection over the Internet. The Internet introduced a new way of breaking files down into much smaller pieces so that they can be downloaded much faster by multiple computers at once. Once all the "pieces" of the file are downloaded, they are put back together on the client's device so that the browser can then view and render the file.

- What code is rendered in the browser?

Webpages are written in HTML. HTML is just a simple text document. However, the text is written in a code, which the programmer writes to tell the browser how they want the browser to display information. When the browser on the client device navigates to a URL, the URL directs to the IP address for the server where the HTML file is stored. The client downloads a temporary copy the HTML file and the browser is then able to read and interpret the code written in the HTML file and display it visually for the user, based on the way the programmer instructed the information to be presented through the code. HTML also typically contains associated code, usually CSS and JavaScript, which enhance the user experience and add additional functionality to the website. CSS adds styling to the HTML code and JavaScript adds interactive functionality like being able to fill out and submit a form directly on the webpage.

- What is the server-side code's main function?

The server side code's main function is to determine what information is sent to the user. The information can be tailored to an individual user. Think Netflix.com, the server-side code determines what movies and TV shows to display based on your user preferences and the items in your watch list.

- What is the client-side code's main function?

The client-side code's main function is to determine how to display the information. Using the example of Netflix.com again, the client-side code tells the browser where on the webpage your watch list will be displayed, or where to show the navigation bar. Where the server-side code is determining what information to display, the client-side code is determining how and where it will be displayed visually on the page.

- What is runtime?

Runtime is the time in which a program or application is running or executing. For example, a runtime error is an error that occurs while the program is running, whereas something like a syntax error would occur before the program is run. 'Compile time' and 'load time' are examples of program states where nothing is being executed and are not considered runtime.

*NOTE
In the interest of being honest, I am doing this because I want to learn and I am very eager to learn, and these final 3 questions really stumped me. Coming in to this process I had a very basic understanding of some HTML concepts but this is a bit more advanced than my current knowledge level. And though I'm usually pretty good at researching and teaching myself new things, even after hours of research I could not find much information available on these topics. So I'm just going to answer them to the best of my knowledge and I look forward to the opportunity to dig further into these topics and more, should I be selected for the apprenticeship.

- How many instances of the client-side assets (HTML, CSS, JS, Images, etc.) are created?

When the client sends a request to the server to get a file, say index.html for example, the file is not saved on the computer, it is only displayed temporarily, and if you were to revisit the same webpage at a later time then a new request for the file would be sent. So no instances of the assets are actually downloaded and stored but during the time in which it is being displayed in the browser, I believe that only one temporary instance is created each time the file is requested by visiting or revisiting the site.

- How many instances of the server-side code are available at any given time?

I believe this would probably vary based on how powerful the server is, but there should be enough instances available to handle any requests that come in. Amazon, for example has millions of visitors every day, therefore they need to have thousands of instances available at any given time.

- How many instances of the databases connected to the server application are created?

I believe it depends on the database but I think some databases create multiple instances while some only create one.